BOEING®

Object Oriented Technology Verification Survey

John Joseph Chilenski

Associate Technical Fellow

Boeing Commercial Airplanes

July 27, 2005

Background

- Boeing is under contract to the FAA to investigate certain issues concerning the verification of object-oriented (software) technology

- The first task (phase) was a survey of the commercial airborne software community for current and anticipated OOT verification practices
  - Conducted 1H 2004

OOTiA Survey

- Survey addressed five areas
  - The use and verification of OOT in commercial aviation in general
    - Not tied to a specific OOTiA development
    - Not tied to specific OOT features or techniques
  - The verification of specific OOT features
  - The confirmation of data coupling and control coupling in both non-OOT and OOT software
  - The structural coverage of source and object code in both non-OOT and OOT software
  - The use and verification of OOTiA in a commercial airborne project
    - Also asked about specific OOT techniques

OOTiA Survey

- Survey was electronically distributed through the FAA’s master software list
    - ~1200 people who have been involved in FAA workshops or conferences

- 10 responses received
    - From 7 of the ~50 OOTiA represented airborne systems and software companies

- Low response due to
    - Low experience level with OOTiA
    - Lack of time

- 11 general findings

OOTiA Survey

1. OOT is already in service in commercial aviation applications, systems or software

- 5 respondents indicated they have OOT software in service
  - 7 systems
    - 2 Level A
    - 4 Level B
    - 1 Level C
- 1 respondent indicated they have a military system with Level A OOT software

OOTiA Survey

2. There will be more usage of OOT in the future

- 6 respondents indicated they had OOT software currently under development, undergoing modification or in the planning stages

BOEING®

OOTiA Survey

3. There are some who are choosing not to use OOT

- 5 respondents indicated this on the survey
- Several indicated this verbally during the personal contacts

OOTiA Survey

4. OOTiA developers are already using the draft OOTiA handbook

- 3 respondents mentioned use of the handbook in their responses

OOTiA Survey

5. Multiple approaches for DO-178B/ED-12B compliance are being used

- Though there were similarities between the responses, all of the respondents had significant differences in their approaches

OOTiA Survey

6. Respondents generally expect that compliance with the objectives of DO-178B/ED-12B is only slightly more difficult for OOT software than for non-OOT software

- Most features of OOT can be mapped onto traditional (non-OOT) features, and handled in a similar fashion

- Note that some respondents took proactive steps to facilitate this

OOTiA Survey

7. Respondents generally expect that verification of OOT software is essentially the same as for non-OOT software

- However, the polymorphism and dynamic binding features of OOT present special challenges

- Where these features are not avoided, which is the general case, special care in their use or verification was indicated

OOTiA Survey

8. Respondents expect that the confirmation of data coupling and control coupling for OOT software will be essentially the same as for non-OOT software

9. Respondents are anticipating no changes to current practices for the confirmation of data coupling and control coupling

- Note: The survey addressed only a part of the mechanism for compliance
  - Process – partly addressed
  - Methods – not addressed
  - Tools – partly addressed

- Note: Effort was not addressed

OOTiA Survey

- Some effort data provided indirectly
  - OOT will take less effort
  - OOT will take more effort

OOTiA Survey

10. Respondents felt that the effectiveness of structural coverage of OOT software was as effective as the structural coverage of non-OOT software

11. Respondents felt that the traceability between source and object codes in OOT software was sufficient enough that object code coverage requirements would be no different than those for non-OOT software

- Note: Two different camps represented
  - Those that believed in subsetting the language and avoiding troublesome constructs
  - Those that believed in mastering an understanding of the behavior of their compiler